APPENDIX XIV

```
FindLinks(V)                                   (* find all physical links associated with Vlan V *)
    If V ≠ unknown then                        (* return links of the server of this Vlan *)
        Let VR be VlanRecord VR corresponding to V.
        Let SR be the server record with SR.Name = VR.ServerName
        Return(SR.Links)
    Else                                       (* for unknown Vlan, return list of all links that have Vlan Servers *)
        LS:= Set Union of SR.Links, taken over all server records SR in ServerList
        Return(LS)

FindVlanPort(V,L,LP)                           (* find Vlan port associated with link port LP on link L *)
    Search through PortMappingList for any entry E with
        E.Vlan = V and E.LinkPort = LP and E.Link = L
    Return (E.VlanPort) or return(NIL) if not found

FindLinkPortofVlan(V,VP,L)                     (* find link port associated with Vlan port VP on link L*)
    Search through PortMappingList for any entry E with
        E.Vlan = V and E.VlanPort = VP and E.Link = L
    Return (E.LinkPort) or return(NIL) if not found

FindLinkPortofProtocol(S,L)                    (* find link port on link L for protocol specifier S*)
    Search through PortMappingList for any entry E with
        E.Specifier = S and and E.Link = L
    Return (E.LinkPort) or return(NIL) if not found

AddPortMapping(V,VP,L,LP,S)                    (* create PortRecord associating Vlan Port VP and link port LP *)
    Search port PortMappingList for any entry E with
        E.Vlan = V, E.VlanPort = VP, E.Link = L, E.LinkPort = LP, and E.Specifier = S
    If not found then add an entry E with E.Vlan = V, E.VlanPort = VP,
        E.LinkPort = P, E.Link = L, E.LinkPort = LP, and E.Specifier = S
```

SourceLookup(SA) (* lookup source address SA in SourceVlanTable, returns a VlanId *)

(* returns UnknownVlanId if unknown and VMLRouterId if SA is a VML Router *)

Split(LinkSet) (* choose one among a set of links, encapsulates splitting policy *)

OpenLinkPort(L) . (*returns a port number on physical link L*)

EnableLinkPort(LP, ProtocolSpecifier)

(*enable receipt of packets matching ProtocolSpecifier on link port LP.ProtocolSpecifier *)

(*abstracts all other detailed info like Protocol Type, SAPs, etc. used in DNA Data Links*)